# The Software Reuse Working Group: A Case Study in Fostering Reuse

Ryan Gerard
*Innovim / NASA Goddard Space Flight Center*
*ryan.gerard@gsfc.nasa.gov*

Robert R. Downs
*Columbia University, Center for International Earth Science Information Network*
*rdowns@ciesin.columbia.edu*

James J. Marshall
*Innovim / NASA Goddard Space Flight Center*
*james.marshall@gsfc.nasa.gov*

Robert E. Wolfe
*NASA Goddard Space Flight Center, Code 614.5*
*robert.e.wolfe@nasa.gov*

## Abstract

*Packaging software assets for reuse can improve the potential for others to adopt the software. Packaging the software with appropriate documentation and other resources can facilitate decision-making by those considering adoption and can enable them to implement the software more efficiently. Software that can be easily integrated is more likely to be shared with others and reused by the recipients.*

*The NASA Earth Science Data Systems (ESDS) Software Reuse Working Group has been chartered to oversee the process that will maximize the reuse of potential software components. As part of this work, a portal web site was created to support reuse practices within the Earth science community. This portal also serves as an example of reuse practices, both as a recipient and as a contributor to reuse.*

*This paper describes the assets that were reused to develop the software reuse portal. For each asset reused, we identify the type, format, and licensing restrictions. This case study exemplifies reuse at various scopes such as the application, component, and function level. Aside from consuming reusable assets, the software produced was also made available for reuse. We address proper packaging methodologies to foster easy integration and reuse.*

*We obtain various reuse metrics based on an analysis of the final system and the impact in our community. We identify caveats in current metrics such as consideration for emerging technologies and intangible benefits.*

## 1. Introduction

The National Aeronautics and Space Administration (NASA) Earth Science Data Systems (ESDS) Software Reuse Working Group (WG) was established in 2004 to promote the reuse of software and related artifacts among members of the ESDS data product and software development community. Members of the ESDS Software Reuse WG include software and data product developers employed by NASA, universities, and other organizations currently working with NASA on grants and contracts.

The Software Reuse WG is chartered [1] to oversee the process that will maximize the reuse potential of software components in order to:

1. Drive down the cost and time of system development, and reduce/eliminate expensive redundancy and duplication in system development,
2. Increase flexibility and responsiveness, and
3. Increase effective and accountable community participation.

The Working Group makes recommendations for activities to NASA Headquarters through the ESDS program office, supports and enables those activities, and also offers recommendations on intellectual property issues.

As part of its initial outreach and education strategy, the ESDS Software Reuse WG began planning for a web presence that would serve both the working group and other members of the community who could benefit by the activities of the working group. A web presence also

1-4244-1500-4/07/$25.00 ©2007 IEEE

could offer community members an opportunity to contribute to working group efforts. The working group decided that the portal should provide a place where members of the community could visit to obtain information on software reuse and to identify other resources that could assist in the reuse of software needed for the development of Earth science data systems.

The working group's outreach and education goals that could be facilitated by a web presence, or portal, were identified. These goals included the development and identification of resources to foster reuse awareness and understanding. Purposes for the portal also were identified [2], including:

- Distribute resources on reuse for the Earth science community,
- Establish a platform for community members to share/exchange resources with each other,
- Be the gateway for reuse information relevant to the community,
- Make access to reuse resources easier, and
- Become the major starting site for reuse within the community.

Based on the purposes identified for the software reuse portal, the working group identified categories of content to populate the portal, including featured projects, asset catalog, reuse tool catalog, reference library, external resources, support tools, hot assets, events, news, and funding opportunities [2]. Each content category was defined and initial mockups were created for the portal content.

The Software Reuse WG then identified the requirements for content management software that could be adopted for the implementation of the software reuse portal. Considering the mission of the Software Reuse WG, open source software was considered. Other requirements identified for the content management software included ease of use and administration capabilities.

After identifying and reviewing candidate content management software packages, the working group proposed the use of Plone for the software reuse portal. Plone is an open source content management system that has been adopted by various organizations for content management applications [3]–[4]. The system administration group reviewed the installation requirements and the operational capabilities of the Plone software to determine whether the software could meet administrative and security requirements for deployment. Plone met these requirements and was selected.

The software reuse portal web site is an information portal, but is not a repository/catalog for reusable assets. The infrastructure and software behind it are not designed for that capability. The Software Reuse WG has recommended that a NASA Earth science software reuse enablement system be developed and has taken several steps in that direction, such as developing the requirements for such a system.

## 2. Software reuse portal development and use

The NASA Software Reuse portal has been developed in stages and continues to be developed. The Plone software was installed on a server for initial testing and review, and accounts were established for the members of the working group support team who would serve as the portal administrators and content providers. The home page of the web site is shown below in Figure 1.

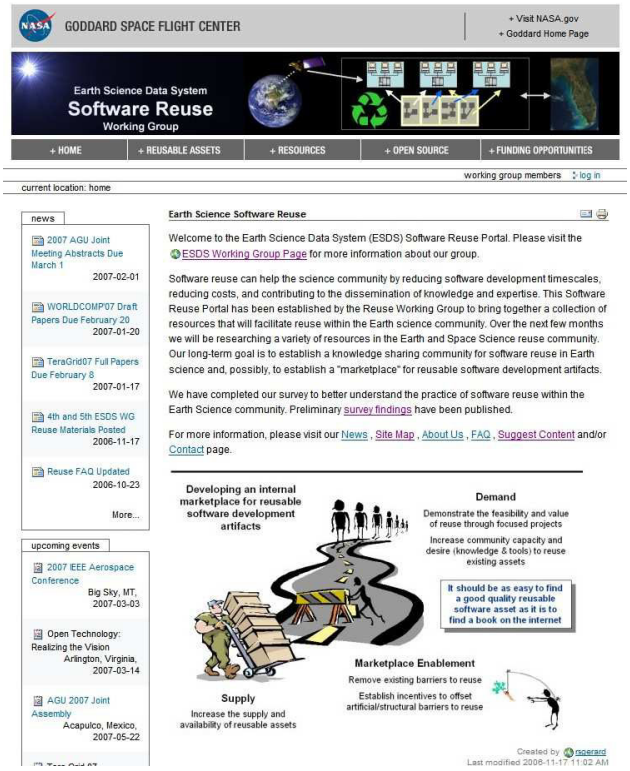


**Figure 1 – Software reuse portal main page (http://www.esdswg.org/softwarereuse/)**

Selected content categories, which had been identified previously, were used to organize the portal for navigation. The working group established criteria and procedures for recommending, reviewing, and selecting items to be included within each content category. In addition, an online form labeled "Suggest a Link" was established to enable visitors to recommend content or submit suggestions for improving the portal. The software reuse portal serves as the gateway for the members of the Software Reuse WG and other members of the Earth science data systems development community who are interested in obtaining information relevant to the reuse of software and other information assets.

During its first two years of operation, the organization of the portal content has evolved and its functionality has been improved to reflect suggestions offered by members of the working group and by other members of the Earth science data systems development community. Statistics are not available for the first six months, but results from November 2005 to February 2007 show that there have been approximately 9300 visits and 31000 page views made by about 7200 unique visitors. The number of visits has been increasing over time, as has the number of repeat visitors. Our site's ranking on the major search engines (Google, Yahoo, and MSN) has been as high as the top three for search terms as general as "software reuse", and the main page earned a Google PageRank of 6 before the site was moved in May 2007.

A number of different software assets were reused in the production of the portal web site. Table 1 provides some basic information about the main components that were reused.

**Table 1 – Key assets reused to make the reuse portal web site**

| Asset Reused | Source | Format | License |
|---|---|---|---|
| Zope (1.3) | Online | Python | ZPL [5] |
| Plone 2.0.5 | Online | Python | GPL [6] |
| NASA CSS/Template | Online (internal) | CSS/HTML | N/A |
| DB Code | Online | DTML | N/A (Example code) |
| Trade Studies | Co-worker webmasters | N/A | N/A |

The Zope Public License (ZPL) "has been certified as open source. It also has been designated as GNU General Public License (GPL) compatible by the Free Software Foundation (FSF)." [5] Zope and Plone build on top of open source software such as Linux, Apache, and Python.

### 2.1. Description of assets reused

Zope is an open source application server written in Python that can be used to build content management systems (CMS) and other similar applications. It makes use of objects stored in a database, rather than files, to publish content to the web. It also provides several options for writing the templates for HTML pages, including Dynamic Template Markup Language (DTML), and Zope Page Templates (ZPT). One CMS that uses Zope is Plone, another open source application. It has a workflow engine and a fine-grained and role-based security model, both of which provide control over the publication of material to the web. It is compliant with standards such as those for accessibility (e.g., Section 508 of the Rehabilitation Act (29 U.S.C. 794d)), and can operate with web services such as the Lightweight Directory Access Protocol (LDAP) or Simple Object Access Protocol (SOAP).

NASA web sites typically have a common look and feel, and NASA helps webmasters produce these sites by supplying them with templates. These templates provide the necessary information to create things like the standard header, footer, layout, color scheme, and menu buttons used on many NASA web sites. With some modifications for use in the Plone environment, we were able to make use of these templates to help simplify the creation of our site.

### 2.2. Forms of software and information reuse

Plone itself demonstrates *application reuse*, which is defined as "reusing an entire application by incorporation of one application inside another" [7]. It is a content management system that "uses the Zope application server infrastructure" [8] as the core of its system. The use of any content management system, where the content of a page can be separated from its appearance/presentation, provides a framework for reuse of a look and feel, such as the One NASA theme used by many NASA sites.

The use of various Plone modules and add-ons could be considered *component level reuse* [7] as they are complete sub-systems utilized to fulfill a requirement. For example, we experimented with a Wiki module developed by a third party that integrates with Plone. This entire component was integrated and evaluated in our system. Though we chose not to use it, this demonstrates reuse at the component level.

A portal feature designed to receive input from the user community and track it in our database was tailored from sample DTML code. The reuse of a software asset for a well-defined function is considered *function reuse* [7]. Our "Suggest a Link" form was created through this type of function reuse.

Other forms of information reuse such as lessons learned came from system maintainers and other web developers who had experience with various content management systems. We utilized these sources for feedback on Plone's ease of use, interoperability, support, flexibility, security, and performance.

## 3. Packaging the portal to foster reuse

Recognizing the needs for two of the other three ESDS working groups to implement portals, the Software Reuse WG created a package to assist any other working groups in the implementation of a web portal using Plone. The package was a compressed zip archive that included the

Plone software, the enhancements that the Software Reuse WG had completed to meet their own needs as well as those needed to meet NASA requirements, and associated documentation. Aspects of the portal design and layout, including the NASA style sheets and the templates for the header and footer, which appear on each page, were also included in the package. The included documentation consisted of a quick start guide, general operating procedures, integration instructions, a reference guide, and source documentation. Packaging the software has enabled other working groups to easily adopt the content management system as a portal, and has enabled others to replicate much of the functionality and appearance that the Software Reuse WG completed during the design, development, and testing of the Software Reuse portal.

As a typical system engineer spends 75% of his/her time on the requirements, design, and debugging process [9], it is critical to include design documentation, design patterns, and/or software architectures in software that is packaged for reuse. In fact, it has been shown that reused requirements, design, and documentation require significantly less debugging [9]. Therefore, the package included documentation to assist other adopters in the installation and implementation process.

## 4. Measuring reuse

Packaging the portal for reuse has offered benefits for the other groups that have adopted various aspects of the software reuse portal to serve their portal needs. Major benefits are saving time and money for development of a portal system. Also, by getting Plone approved for use at Goddard Space Fight Center and creating a NASA template for Plone web sites, the Software Reuse WG has made it easier for other groups to run their own Plone web sites.

The savings of time and money of each instance of reuse can be approximated using existing formulas. For example, the COCOMO (Constructive Cost Modeling) model developed by Barry Boehm [10] provides a method for estimating the cost, effort, and schedule involved in software development activities. For our purposes here, we have made the estimate using COCOMO 81, the original version, which provides a relatively simple way to estimate effort. Additional information can be found at, for example, the Center for Software Engineering's web page [11].

The basic concept of the COCOMO 81 model is that the development effort can be expressed as $E=a\times S^b$, with effort E in staff-months and size S in thousands of lines of code (KLOC). The factors a and b are constants that change according to the estimate required. Projects are categorized as organic, semi-detached, and embedded, primarily by their size. Typically, organic projects are small, semi-detached projects are of medium size, and embedded projects are large. For the basic COCOMO model used here, the factors a and b for each size/type project are given in Table 2 below.

**Table 2 – Factors a and b for COCOMO 81**

| | Organic | Semi-detached | Embedded |
|---|---|---|---|
| a | 2.4 | 3.0 | 3.6 |
| b | 1.05 | 1.12 | 1.20 |

We estimate that we have written approximately 500 new lines of code (S=0.500), and that the project is semi-detached (a=3.0 and b=1.12). This corresponds to a development effort of approximately 1.4 staff-months. This is a very conservative estimate, as it does not include the effort to learn emerging and/or relatively obscure technologies such as DTML and the Gadfly database, the relational database management system written in Python that is provided with Zope. We have provided three other groups with the results of our work, in order to help them produce Plone web sites that conform to the NASA look and feel. By our previous calculation, each group has saved 1.4 staff-months of development time by utilizing our software packaged for reuse. This translates to a total savings of 4.2 staff-months, which accounts for savings to date. More recently, we have been approached by other groups seeking our work. If they choose to reuse what we have done, the total savings will be higher than our current estimate.

One can also use a general formula to approximate percentage of reuse [12]:

$$\text{Reuse } \% = (\textit{Reused Statements} / \textit{Total Statements}) \times 100$$

Using this formula and a conservative estimate for the number of lines used from Plone, Zope, NASA templates, etc., approximately 95% of our system was created through reuse.

The use of Plone and Zope themselves save development effort, since the functions they provide do not have to be developed from scratch. The Software Reuse WG determined that Plone and Zope meet most of the functional requirements for our portal web site, and would presumably do so for any other group choosing to use them, so we use many of the features provided by these software packages. This reuse has noticeably reduced our development effort, and the software reuse portal web site was produced in about 6-8 weeks. Having learned how to install and configure a Plone web site, we can now assist others in producing a similar site in about 4 hours [13].

However, the benefits go beyond that of code reuse. In order to run a web site using the Plone content

management system, we had to get approval from system administrators and other relevant parties, since this involved installing a new package on our web server. As the first web site on that web server, we also had to follow specific procedures to enable public access to the web site. Some other groups who use the same server have benefited from these efforts, since they do not have to go through all of these processes. Plone has been accepted for use and the web server is publicly accessible, creating additional reuse-related savings.

## 5. Discussion, lessons learned, and next steps

Members of the working group also provided assistance to other groups by answering questions about the implementation and use of Plone and the enhancements that have been packaged with it. Answering such questions has provided an opportunity to consider improvements. For example, we were able to improve the workflow configuration of our own site after helping another ESDS working group with their site's workflow. Also, this has made it easier for these groups to implement their sites since they can reuse the knowledge and experience of the Software Reuse WG.

The server that hosts the software reuse portal has been taken off line on a few occasions due to maintenance requirements and security concerns. Other portals, in addition to the software reuse portal, are being hosted on the same server, which has increased the necessity to take the server off line. Because of this, we recommend that the adopters of Plone fully implement and test their portals on a separate development environment before deploying on a production server that is being used to support communities. We also believe that improvements in the documentation to support deployment could help avoid or lessen such problems. We have learned that more communication, on a more regular basis, with the groups hosting the other portals is a way to ensure that we are all making the best use of Plone possible.

One of the lessons learned from our experience with Plone is how to manage users and privileges. This is an important part of the configuration that helps with the security of the site. Privileges can be set at the user level so that, for example, certain users are only allowed to edit certain sections of the site, and no one (except the administrator) can edit everything. Also, content can be put into different states, such as public or private. Who has access to private content is part of the user privileges, and can only be given to people with valid user accounts on the Plone system. This allows us to do things such as place content on the site and work on it before it goes public by leaving it in a private state, or to remove some older material that may be less relevant by putting it into a private state instead of deleting it from the site.

Another lesson learned deals with workflow management. This is another configuration that affects the security of the site. Our site has been configured so that no changes can go live on the site without the administrator's approval. The workflow we have set up requires users to submit their changes for approval before they are made to the live site. This gives the administrator the opportunity to check all edits and confirm that they are relevant and necessary before allowing them to be made public. By using a well-defined workflow process, it is clear what state content is in, and how it is permitted to changes states (e.g., to become publicly viewable).

While some of these lessons may appear to be specific to the use of Plone, they are applicable to any other system that has a similar structure. Our lesson on managing users and privileges can apply to any system that has a similar method of granting privileges to users. Our lesson on workflow can be used with any system that has a similar workflow process.

We have also noticed that software reused in other applications tends to market itself. In our case study, some of the projects who reused our web template received inquiries relating to reusing their software. These projects can direct people back to us as the original creators of the template. However, if they choose to pass on the work they reused instead, we have a second generation of reuse: the reuse of reused work. This is evidence that reuse can grow beyond what the original creators are aware of, resulting in higher total savings than estimated by the creators.

Our next steps for the site include providing additional resources on reuse recommended by community members. For example, working group members have created guidelines on various reuse-related topics from the developer's perspective (bottom-up reuse) in response to feedback that previous work by the working group had a more top-down perspective. We are also working on developing a reuse enablement system for the community of Earth science software developers in order to help them share reusable software assets more easily. Our goal is to incorporate this system into the portal web site to provide easy access to the information it contains in a location already familiar to current users.

## 6. Summary and conclusions

While this is a medium scale example of software reuse, the key elements to fostering reuse apply in any scope. Regardless of the level of reuse (application reuse, component reuse, functional reuse, etc.), clear documentation, and often support from previous adopters, are necessary to aid adoption. Packaging such supporting materials along with the software has enabled others to

adopt the software with much less effort than that experienced by the Software Reuse WG [13].

While this case study demonstrates value-added aspects of packaging software to improve the potential for software reuse, the existence of other barriers to software reuse also are recognized. For example, organizational aspects also can influence the potential for successful software reuse [14].

It is also important to note the significant savings realized from making a software asset available for reuse. While it can be hard to accurately quantify the benefit from reuse, long-term analysis of cost-benefits become evident over time.

## 7. Acknowledgments

The authors would like to acknowledge the funding and support received from NASA, especially the support from Kathy Fontaine, the Earth Science Data System Working Group Coordinator; the support from NASA contract NAS5-03117 for Robert Downs; and the support from NASA for other members of the Software Reuse WG.

## 8. References

[1] NASA ESDS Software Reuse Working Group, "Charter for the SEEDS Reuse Working Group", 2004.

[2] Downs, R.R., Giles, D.M., and Olding, S. "Reuse Portal Planning to Support Reuse by the Scientific Community", poster presentation, 2nd Earth Science Data Systems Joint Working Group Meeting, October 18, 2004.

[3] Adams, P., Boldyreff, C., Nutter, D., and Rank, S. 2005. Adaptive reuse of libre software systems for supporting on-line collaboration. *SIGSOFT Softw. Eng. Notes* 30, 4 (Jul. 2005), 1-4.

[4] Adams, P., Boldyreff, C., Nutter, D., and Rank, S. 2005. Adaptive reuse of libre software systems for supporting on-line collaboration. In *Proceedings of the Fifth Workshop on Open Source Software Engineering* (St. Louis, Missouri, May 17 - 17, 2005). 5-WOSSE. ACM Press, New York, NY, 1-4.

[5] Zope Corporation. (2006). Zope Public License. [Online]. Available web site: http://www.zope.org/Resources/License/.

[6] Limi, Alexander. (2006, June). Plone License FAQ. [Online]. Available web site: http://plone.org/about/copyrights/license-faq/

[7] Maxim, B.R. (2001, March). Software Reuse and Component-Based Software Engineering. [Online]. Available web site: http://www.engin.umd.umich.edu/CIS/course.des/cis376/ppt/lec22.ppt

[8] Thiruvathukal, G.K.; Laufer, K. Plone and Content Management. Computing in Science & Engineering, Vol.06, Iss.4, July-Aug.2004 Pages: 88-95.

[9] Anderson, K.M., (2004, September). Lecture 10: Software Re-use. [Online]. Available web site: http://www.cs.colorado.edu/~kena/classes/3308/f04/lectures/lecture10.pdf

[10] Boehm, Barry W., Software Engineering Economics, Prentice Hall, 1981.

[11] University of Southern California Center for Software Engineering, COCOMO. [Online]. Available web site: http://sunset.usc.edu/research/COCOMOII/index.html

[12] Poulin, Jeffrey S., "Software Reuse Metrics, Reusability Metrics, and Economic Models: A Tutorial," *Proceedings of the Seventh Annual Software Technology Conference (STC'96),* Salt Lake City, Utah, 21-26 April 1996.

[13] Gerard, R., Downs, R.R., Marshall, J.J., Wolfe, R.E. (2006), The Software Reuse Portal: A Case Study in Packaging Software to Contribute to Reuse Practices, Eos Trans. AGU, 87(52), Fall Meet. Suppl., Abstract IN13D-1176

[14] Sherif, Karma, Vinze, Ajay. (2003). Barriers to adoption of software reuse: a qualitative study. Information & Management, 41(2), p. 159-175, December 2003. doi:10.1016/S0378-7206(03)00045-4